Indian Buffet process for model selection in convolved multiple-output Gaussian processes

Cristian Guarnizo, Mauricio A. Álvarez

Faculty of Engineering, Universidad Tecnológica de Pereira, Pereira, Colombia.

Abstract

Multi-output Gaussian processes have received increasing attention during the last few years as a natural mechanism to extend the powerful flexibility of Gaussian processes to the setup of multiple output variables. The key point here is the ability to design kernel functions that allow exploiting the correlations between the outputs while fulfilling the positive definiteness requisite for the covariance function. Alternatives to construct these covariance functions are the linear model of coregionalization and process convolutions. Each of these methods demand the specification of the number of latent Gaussian process used to build the covariance function for the outputs. We propose in this paper, the use of an Indian Buffet process as a way to perform model selection over the number of latent Gaussian processes. This type of model is particularly important in the context of latent force models, where the latent forces are associated to physical quantities like protein profiles or latent forces in mechanical systems. We use variational inference to estimate posterior distributions over the variables involved, and show examples of the model performance over artificial data, a motion capture dataset, and a gene expression dataset.

1 Introduction

Kernel methods for vector-valued functions have proved to be an important tool for designing learning algorithms that perform multi-variate regression (Bonilla et al., 2008), and multi-class classification (Skolidis and Sanguinetti, 2011). A kernel function that encodes suitable correlations between output variables can be embedded in established machine learning algorithms like support vector machines or Gaussian process predictors, where the kernel function is interpreted as a covariance function.

Different kernels for vector-valued functions proposed in recent years within the machine learning community, are particular cases of the so called linear model of coregionalization (LMC) (Journel and Huijbregts, 1978; Goovaerts, 1997), heavily used for cokriging in geostatistics (Chilès and Delfiner, 1999; Cressie, 1993). Furthermore, the linear model of coregionalization turns out to be a special case of the so called process convolutions (PC) used in statistics for developing covariance functions (Ver Hoef and Barry, 1998; Higdon, 1998). For details, see Álvarez et al. (2012).

Under the LMC or the PC frameworks, the way in which a kernel function for multiple variables is constructed, follows a similar pattern: a set of orthogonal Gaussian processes, each of them characterized by an specific covariance function, are linearly combined to represent each of the output variables. Typically, each of these Gaussian processes establishes the degree of smoothness that is to be explained in the outputs. In PC, the set of orthogonal Gaussian processes are initially smoothed through a convolution operation that involves the specification of the so called smoothing kernels. The smoothing kernel may be the impulse response of a dynamical system, or, in general, may correspond to the Green's function associated to a differential equation. Gaussian processes that use a kernel constructed from a PC with a Green's function as a smoothing kernel, have been coined by the authors of Álvarez et al. (2009) as latent force models.

Despite its success for prediction, it is still unclear how to select the number of orthogonal Gaussian processes used for building the multi-output Gaussian process or a latent force model. Furthermore, in the context of latent force models where these orthogonal Gaussian processes may represent a physical quantity, like the action of a protein for transcription regulation of a gene or a latent force in a system involving masses and dampers, it

becomes relevant to undercover the interactions between the latent Gaussian processes and the output variables that are being modelled.
In this paper, we use an Indian Buffet Process (IBP) (Griffiths and Ghahramani, 2005, 2011) for model selection in convolved multiple output Gaussian processes. The IBP is a non-parametric prior over binary matrices, that imposes an structure over the sparsity pattern of the binary matrix. It has previously been used for introducing sparsity in linear models (Knowles and Ghahramani, 2011). We formulate a variational inference procedure for inferring posterior distributions over the structure of the relationships between output functions and latent processes, by combining ideas from Álvarez et al. (2010) and Doshi-Velez et al. (2009). We show examples of the model using artificial data, motion capture data and a gene expression dataset.

2 Convolved multiple output Gaussian processes

We want to jointly model D output functions $\{f_d(\mathbf{x})\}_{d=1}^D$, where each output $f_d(\mathbf{x})$ can be written as

$$f_d(\mathbf{x}) = \sum_{q=1}^{Q} S_{d,q} \int_{\mathcal{X}} G_d(\mathbf{x} - \mathbf{x}') u_q(\mathbf{x}') \, \mathrm{d}\, \mathbf{x}', \tag{1}$$

where $G_d(\mathbf{x} - \mathbf{x}')$ are smoothing functions or smoothing kernels, $\{u_q(\mathbf{x})\}_{q=1}^Q$ are orthogonal processes, and the variables $\{S_{d,q}\}_{d=1,q=1}^{D,Q}$ measure the influence of the latent function q over the output function d. We assume that each latent process $u_q(\mathbf{x})$ is a Gaussian process with zero mean function and covariance function $k_q(\mathbf{x}, \mathbf{x}')$.
The model above is known in the geostatistics literature as a process convolution. If $G_d(\cdot)$ is equal to the Dirac delta function, then the linear model of coregionalization is recovered (Álvarez et al., 2012). Also, in the context of linear dynamical systems, the function $G_d(\cdot)$ is related to the so called impulse response of the system.

2.1 Covariance functions

Due to the linearity in expression (1), the set of processes $\{f_d(\mathbf{x})\}_{d=1}^D$ follow a joint Gaussian process with mean function equal to zero, and covariance function given by

$$k_{f_d, f_{d'}}(\mathbf{x}, \mathbf{x}') = \mathrm{cov}\left[f_d(\mathbf{x}) f_{d'}(\mathbf{x}')\right] = \sum_{q=1}^{Q} S_{d,q} S_{d',q} k_{f_d^q, f_{d'}^q}(\mathbf{x}, \mathbf{x}'),$$

where we have defined

$$k_{f_d^q, f_{d'}^q}(\mathbf{x}, \mathbf{x}') = \int_{\mathcal{X}} \int_{\mathcal{X}} G_d(\mathbf{x} - \mathbf{z}) G_{d'}(\mathbf{x} - \mathbf{z}') k_q(\mathbf{z}, \mathbf{z}') \, \mathrm{d}\, \mathbf{z} \, \mathrm{d}\, \mathbf{z}'. \tag{2}$$

Besides the covariance function defined above, we are interested in the covariance function between $f_d(\mathbf{x})$ and $u_q(\mathbf{x})$, which follows

$$k_{f_d, u_q}(\mathbf{x}, \mathbf{x}') = \mathrm{cov}\left[f_d(\mathbf{x}) u_q(\mathbf{x}')\right] = S_{d,q} \int_{\mathcal{X}} G_d(\mathbf{x} - \mathbf{z}) k_q(\mathbf{z}, \mathbf{x}') \, \mathrm{d}\, \mathbf{z}. \tag{3}$$

For some forms of the smoothing kernel $G_d(\cdot)$, and the covariance function $k_q(\cdot)$, the covariance functions $k_{f_d^q, f_{d'}^q}(\mathbf{x}, \mathbf{x}')$ and $k_{f_d, u_q}(\mathbf{x}, \mathbf{x}')$ can be worked out analytically. We show some examples in section 6.1.

2.2 Likelihood model for multi-output regression

In a multi-variate regression setting the likelihood model for each output can be expressed as

$$y_d(\mathbf{x}) = f_d(\mathbf{x}) + w_d(\mathbf{x}),$$

where each $f_d(\mathbf{x})$ is given by (1), and $\{w_d(\mathbf{x})\}_{d=1}^D$ are a set of processes that could represent a noise process for each output. Assuming that each $w_d(\mathbf{x})$ is also a Gaussian process with zero mean and covariance function $k_{w_d, w_d}(\mathbf{x}, \mathbf{x}')$, the covariance function between $y_d(\mathbf{x})$, and $y_{d'}(\mathbf{x}')$ is given by

$$k_{y_d, y_{d'}}(\mathbf{x}, \mathbf{x}') = k_{f_d, f_{d'}}(\mathbf{x}, \mathbf{x}') + k_{w_d, w_d}(\mathbf{x}, \mathbf{x}') \delta_{d,d'},$$

where $\delta_{d,d'}$ is the Kronecker delta.

2.3 Inference and hyper-parameter learning

Let $\mathcal{D} = \{\mathbf{X}_d, \mathbf{y}_d\}_{d=1}^D$ be a dataset for a multi-output regression problem. We use $\mathbf{X}$ to refer to the set $\{\mathbf{X}_d\}_{d=1}^D$, and $\mathbf{y}$ to refer to the set $\{\mathbf{y}_d\}_{d=1}^D$. We assume that we have N data observations for each output. The posterior distribution $p(\mathbf{u}|\mathbf{y})$, and the predictive distribution for $\mathbf{f}_*$ at a new input point $\mathbf{x}_*$ can both be computed using standard Gaussian processes formulae (Rasmussen and Williams, 2006).
Different methods have been proposed for performing computationally efficient inference and hyperparameter learning in multi-output Gaussian processes (Álvarez and Lawrence, 2011), reducing the computational complexity from $\mathcal{O}(N^3D^3)$ to $\mathcal{O}(NDM^2)$, where M is a user-specified value.

3 The Indian Buffet process

An open question in models like the one described in Eq. (1) is how to choose the number of latent functions Q. In this report, we will use an Indian Buffet process as a prior to automatically choose Q.
The IBP is a distribution over binary matrices with a finite number of rows and an unbounded number of columns (Griffiths and Ghahramani, 2005). This can define a non-parametric latent feature model in which rows are related to data points and columns are related to latent features. The relationship between latent features and data points can be encoded in a binary matrix $\mathbf{Z} \in \mathbb{R}^{D\times Q}$, where $Z_{d,q} = 1$ if feature q is used to explain data point d and $Z_{d,q} = 0$ otherwise. Each element $Z_{d,q}$ of the matrix $\mathbf{Z}$ is sampled as follows

$$\begin{aligned} v_j &\sim \text{Beta}(\alpha, 1), \\ \pi_q &= \prod_{j=1}^{q} v_j, \\ Z_{d,q} &\sim \text{Bernoulli}(\pi_q), \end{aligned}$$

where α is a real positive value, and π_q is the probability of observing a non-zero value in the column q of the matrix $\mathbf{Z}$, this is, the value π_q controls the sparsity for the latent feature q. As we will see, in our proposed model, the value of α is related to the average number of latent functions per output.
Using an IBP as a prior for a linear Gaussian model, the authors in Doshi-Velez et al. (2009), derive two variational mean field approximations, referred to as a "finite variational approach", and an "infinite variational approach". We adopt the latter approach, this is, even though that the update equations will be based on the true IBP posterior over an infinite number of features, for a practical implementation, we use a level of truncation Q as the maximum number of latent functions. In this approach, as shown in previous equations, $\boldsymbol{\upsilon} = \{v_j\}_{j=1}^Q$ are independent samples from a Beta distribution, while $\boldsymbol{\pi} = \{\pi_q\}_{q=1}^Q$ are dependent variables obtained by multiplying the sampled values for v_j as shown before. Thus, in the factorised variational distribution of our mean field approach we use $\boldsymbol{\upsilon}$ as hidden variables with the prior given before. We induce sparsity over the sensitivities by pre-multiplying $Z_{d,q}$ with $S_{d,q}$, as explained in the next section.

4 Variational formulation for model selection

The model selection approach presented here is based on the variational formulation for convolved multiple output Gaussian processes proposed by Álvarez et al. (2010), and the variational formulation for the Indian Buffet Process proposed by Doshi-Velez et al. (2009). We start by defining the likelihood as

$$p(\mathbf{y}|u, \mathbf{X}, \boldsymbol{\theta}, \mathbf{S}, \mathbf{Z}) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \boldsymbol{\Sigma}_{\mathbf{w}}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{y}_d|\mathbf{f}_d, \boldsymbol{\Sigma}_{\mathbf{w}_d}),$$

where $u = \{u_q\}_{q=1}^Q$, $\mathbf{S} = [S_{d,q}] \in \Re^{D\times Q}$, $\mathbf{Z} = [Z_{d,q}] \in \{0,1\}^{D\times Q}$ and each output vector $\mathbf{f}_d$ is defined as

$$\begin{bmatrix} f_d(\mathbf{x}_1) \\ f_d(\mathbf{x}_2) \\ \vdots \\ f_d(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} \sum_{q=1}^Q Z_{d,q} S_{d,q} \int_{\mathcal{X}} G_{d,q}(\mathbf{x}_1 - \mathbf{z}) u_q(\mathbf{z}) \mathrm{d}\mathbf{z} \\ \sum_{q=1}^Q Z_{d,q} S_{d,q} \int_{\mathcal{X}} G_{d,q}(\mathbf{x}_2 - \mathbf{z}) u_q(\mathbf{z}) \mathrm{d}\mathbf{z} \\ \vdots \\ \sum_{q=1}^Q Z_{d,q} S_{d,q} \int_{\mathcal{X}} G_{d,q}(\mathbf{x}_N - \mathbf{z}) u_q(\mathbf{z}) \mathrm{d}\mathbf{z} \end{bmatrix}.$$

For each latent function $u_q(\cdot)$, we define a set of auxiliary variables or inducing variables $\mathbf{u}_q \in \mathbb{R}$, obtained when evaluating the latent function u_q at a set of M inducing inputs $\{\mathbf{z}_m\}_{m=1}^M$. We refer to the set of inducing variables using $\mathbf{u} = \{\mathbf{u}_q\}_{q=1}^Q$. Following ideas used in several computationally efficient Gaussian process methods, we work with the conditional densities $p(u|\mathbf{u})$, instead of the full Gaussian process $p(u)$. The conditional density of the latent functions given the inducing variables can be written as

$$p(u|\mathbf{u}) = \prod_{q=1}^Q \mathcal{N}(u_q|\mathbf{k}_{u_q,\mathbf{u}_q}^\top \mathbf{K}_{\mathbf{u}_q,\mathbf{u}_q}^{-1} \mathbf{u}_q, k_{u_q,u_q} - \mathbf{k}_{u_q,\mathbf{u}_q}^\top \mathbf{K}_{\mathbf{u}_q,\mathbf{u}_q}^{-1} \mathbf{k}_{u_q,\mathbf{u}_q}),$$

with $\mathbf{k}_{u_q,\mathbf{u}_q}^\top = [k_{u_q,u_q}(\mathbf{z},\mathbf{z}_1), k_{u_q,u_q}(\mathbf{z},\mathbf{z}_2), \ldots, k_{u_q,u_q}(\mathbf{z},\mathbf{z}_M)]$. The prior over $\mathbf{u}$ has the following form

$$p(\mathbf{u}) = \prod_{q=1}^Q \mathcal{N}(\mathbf{u}_q|\mathbf{0}, \mathbf{K}_{\mathbf{u}_q,\mathbf{u}_q}).$$

For the elements of $\mathbf{S}$ we use an spike and slab prior as follows (Knowles and Ghahramani, 2011)

$$p(S_{d,q}|Z_{d,q},\gamma_{d,q}) = Z_{d,q}\mathcal{N}(S_{d,q}|0,\gamma_{d,q}^{-1}) + (1 - Z_{d,q})\delta(S_{d,q}),$$

where $Z_{d,q}$ are the elements of the binary matrix $\mathbf{Z}$ that follows an Indian Buffet Process Prior. This is different from Titsias and Lázaro-Gredilla (2011) where all the variables $Z_{d,q}$ are drawn from the same Bernoulli distribution with parameter π. From the previous section we know that the prior for $Z_{d,q}$ is given by

$$p(Z_{d,q}|\pi_q) = \text{Bernoulli}(Z_{d,q}|\pi_q).$$

To apply the variational method, we write the joint distribution for $Z_{d,q}$ and $S_{d,q}$ as

$$\begin{aligned} p(S_{d,q}, Z_{d,q}|\pi_q,\gamma_{d,q}) =& p(S_{d,q}|Z_{d,q},\gamma_{d,q})p(Z_{d,q}|\pi_q) \\ =& [\pi_q \mathcal{N}(S_{d,q}|0,\gamma_{d,q}^{-1})]^{Z_{d,q}} [(1-\pi_q)\delta(S_{d,q})]^{1-Z_{d,q}}, \end{aligned}$$

leading to

$$p(\mathbf{S},\mathbf{Z}|\boldsymbol{\upsilon},\boldsymbol{\gamma}) = \prod_{d=1}^D \prod_{q=1}^Q [\pi_q \mathcal{N}(S_{d,q}|0,\gamma_{d,q}^{-1})]^{Z_{d,q}} [(1-\pi_q)\delta(S_{d,q})]^{1-Z_{d,q}}.$$

We assume that the hyperparameters $\gamma_{d,q}$ follow a Gamma prior,

$$p(\boldsymbol{\gamma}) = \prod_{d=1}^D \prod_{q=1}^Q \text{Gamma}(\gamma_{d,q}|a_{d,q}^\gamma, b_{d,q}^\gamma).$$

Although not written explicitly, the idea here is that Q can be as high as we want to. In fact, the IBP prior assumes that $Q \to \infty$, but in our variational inference implementations the value of Q is fixed and it represents the truncation level on the IBP (see Doshi-Velez et al. (2009)). According to our model, the complete likelihood follows as

$$p(\mathbf{y},\mathbf{X},u,\mathbf{S},\mathbf{Z},\boldsymbol{\upsilon},\boldsymbol{\gamma},\boldsymbol{\theta}) = p(\mathbf{y}|\mathbf{X},u,\mathbf{S},\mathbf{Z},\boldsymbol{\upsilon},\boldsymbol{\gamma},\boldsymbol{\theta})p(u|\boldsymbol{\theta})p(\mathbf{S},\mathbf{Z}|\boldsymbol{\gamma},\boldsymbol{\upsilon})p(\boldsymbol{\upsilon})p(\boldsymbol{\gamma}),$$

where $\boldsymbol{\theta}$ are the hyperparameters regarding the type of covariance function (see section 6.1). For the variational distribution, we use a mean field approximation, and assume that the terms in the posterior factorize as

$$q(\mathbf{u}) = \prod_{q=1}^{Q} q(\mathbf{u}_q), \quad q(\mathbf{S}, \mathbf{Z}) = \prod_{d=1}^{D}\prod_{q=1}^{Q} q(S_{d,q}|Z_{d,q})q(Z_{d,q}), \quad q(\boldsymbol{\gamma}) = \prod_{d=1}^{D}\prod_{q=1}^{Q} q(\gamma_{d,q}), \quad q(\boldsymbol{\upsilon}) = \prod_{q=1}^{Q} q(\upsilon_q).$$

Following the same formulation used by Álvarez et al. (2009), the posterior takes the form

$$q(u, \mathbf{u}, \mathbf{S}, \mathbf{Z}, \boldsymbol{\upsilon}, \boldsymbol{\gamma}) = p(u|\mathbf{u})q(\mathbf{u})q(\mathbf{S}, \mathbf{Z})q(\boldsymbol{\gamma})q(\boldsymbol{\upsilon}).$$

The lower bound that needs to be maximized, $F_V(q(\mathbf{u}), q(\mathbf{S}, \mathbf{Z}), q(\boldsymbol{\gamma}), q(\boldsymbol{\upsilon}))$, is given as (Bishop, 2006)

$$\int q(u, \mathbf{u}, \mathbf{S}, \mathbf{Z}, \boldsymbol{\upsilon}, \boldsymbol{\gamma}) \log \left\{ \frac{p(\mathbf{y}|\mathbf{X}, u, \mathbf{S}, \mathbf{Z}, \boldsymbol{\upsilon}, \boldsymbol{\gamma}, \boldsymbol{\theta})p(u|\mathbf{u})p(\mathbf{u})p(\mathbf{S}, \mathbf{Z}|\boldsymbol{\upsilon}, \boldsymbol{\gamma})p(\boldsymbol{\upsilon})p(\boldsymbol{\gamma})}{q(u, \mathbf{u}, \mathbf{S}, \mathbf{Z}, \boldsymbol{\upsilon}, \boldsymbol{\gamma})} \right\} \mathrm{d}u\, \mathrm{d}\mathbf{u}\, \mathrm{d}\mathbf{S}\, \mathrm{d}\mathbf{Z}\, \mathrm{d}\boldsymbol{\upsilon}\, \mathrm{d}\boldsymbol{\gamma}.$$

By using standard variational equations (Bishop, 2006), it can be shown that the lower bound F_V is given as

$$\begin{aligned}
F_V = {} & \frac{1}{2}\mathbf{y}^\top \boldsymbol{\Sigma}_{\mathbf{w}}^{-1} \widetilde{\mathbf{K}}_{\mathbf{f},\mathbf{u}} \widetilde{\mathbf{A}}^{-1} \widetilde{\mathbf{K}}_{\mathbf{f},\mathbf{u}}^\top \boldsymbol{\Sigma}_{\mathbf{w}}^{-1} \mathbf{y} - \frac{1}{2}\log|\widetilde{\mathbf{A}}| + \frac{1}{2}\log|\mathbf{K}_{\mathbf{u},\mathbf{u}}| - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\operatorname{tr}(\boldsymbol{\Sigma}_{\mathbf{w}}^{-1}\mathbf{y}\mathbf{y}^\top) \\
& - \frac{1}{2}\sum_{d=1}^{D}\sum_{q=1}^{Q} \mathrm{E}[Z_{d,q}S_{d,q}^2] \operatorname{tr}\left[\boldsymbol{\Sigma}_{\mathbf{w}_d}^{-1}\mathbf{K}_{\mathbf{f}_d|\mathbf{u}_q}\right] - \frac{1}{2}\log 2\pi \sum_{d=1}^{D}\sum_{q=1}^{Q} \mathrm{E}[Z_{d,q}] + \sum_{d=1}^{D}\sum_{q=1}^{Q} \mathrm{E}[Z_{d,q}]\,\mathrm{E}[\log \pi_q] \\
& + \frac{1}{2}\sum_{d=1}^{D}\sum_{q=1}^{Q} \mathrm{E}[Z_{d,q}]\left[\psi(a_{d,q}^{\gamma^*}) - \log b_{d,q}^{\gamma^*}\right] - \frac{1}{2}\sum_{d=1}^{D}\sum_{q=1}^{Q} \frac{a_{d,q}^{\gamma^*}}{b_{d,q}^{\gamma^*}} \mathrm{E}[Z_{d,q}S_{d,q}^2] \\
& + \sum_{d=1}^{D}\sum_{q=1}^{Q} (1 - \mathrm{E}[Z_{d,q}])\,\mathrm{E}[\log(1-\pi_q)] - \sum_{d=1}^{D}\sum_{q=1}^{Q} \log\Gamma(a_{d,q}^{\gamma}) + \sum_{d=1}^{D}\sum_{q=1}^{Q} a_{d,q}^{\gamma}\log b_{d,q}^{\gamma} \\
& + \sum_{d=1}^{D}\sum_{q=1}^{Q} \left(a_{d,q}^{\gamma} - 1\right)\left[\psi(a_{d,q}^{\gamma^*}) - \log b_{d,q}^{\gamma^*}\right] - \sum_{d=1}^{D}\sum_{q=1}^{Q} b_{d,q}^{\gamma}\frac{a_{d,q}^{\gamma^*}}{b_{d,q}^{\gamma^*}} + (\alpha - 1)\sum_{q=1}^{Q}[\psi(\tau_{q1}) - \psi(\tau_{q1} + \tau_{q2})] \\
& + \sum_{d=1}^{D}\sum_{q=1}^{Q} \eta_{d,q}\left[\frac{1}{2}\log v_{d,q} + \frac{1}{2}(1 + \log(2\pi))\right] + \sum_{d=1}^{D}\sum_{q=1}^{Q}[-\eta_{d,q}\log\eta_{d,q} - (1-\eta_{d,q})\log(1-\eta_{d,q})] \\
& + \sum_{q=1}^{Q}\left[\log\left(\frac{\Gamma(\tau_{q1})\Gamma(\tau_{q2})}{\Gamma(\tau_{q1}+\tau_{q2})}\right) - (\tau_{q1}-1)\psi(\tau_{q1}) - (\tau_{q2}-1)\psi(\tau_{q2}) + (\tau_{q1}+\tau_{q2}-2)\psi(\tau_{q1}+\tau_{q2})\right] \\
& + \sum_{d=1}^{D}\sum_{q=1}^{Q}\left[\log\Gamma(a_{d,q}^{\gamma^*}) - (a_{d,q}^{\gamma^*} - 1)\psi(a_{d,q}^{\gamma^*}) - \log b_{d,q}^{\gamma^*} + a_{d,q}^{\gamma^*}\right],
\end{aligned}$$

where $\widetilde{\mathbf{K}}_{\mathbf{f},\mathbf{u}} = \mathbf{E}_{\mathbf{ZS}} \odot \mathbf{K}_{\mathbf{f},\mathbf{u}}$ and $\widetilde{\mathbf{A}} = \mathbf{A} + \widetilde{\mathbf{K}}_{\mathbf{f},\mathbf{u}}^\top \boldsymbol{\Sigma}_{\mathbf{w}}^{-1} \widetilde{\mathbf{K}}_{\mathbf{f},\mathbf{u}}$. Besides, $\mathbf{E}_{\mathbf{ZS}}$ is a block-wise matrix with blocks $\mathrm{E}[Z_{d,q}S_{d,q}]\mathbf{1}_{N\times N}$, $\mathbf{A} = \mathbf{K}_{\mathbf{u},\mathbf{u}} + \overline{\mathbf{K}}_{\mathbf{u},\mathbf{u}}$, $\overline{\mathbf{K}}_{\mathbf{u},\mathbf{u}} = \mathbf{M} \odot \left(\widehat{\mathbf{K}}_{\mathbf{f},\mathbf{u}}^\top \boldsymbol{\Sigma}_{\mathbf{w}}^{-1} \widehat{\mathbf{K}}_{\mathbf{f},\mathbf{u}}\right)$, with $\mathbf{M}$ being a block-diagonal matrix with blocks $\mathbf{1}_{N\times N}$, and $\widehat{\mathbf{K}}_{\mathbf{f},\mathbf{u}} = \mathbf{V}_{\mathbf{ZS}} \odot \mathbf{K}_{\mathbf{f},\mathbf{u}}$, with $\mathbf{V}_{\mathbf{ZS}}$ being a block-wise matrix with blocks given by $\left(\mathrm{E}[Z_{d,q}S_{d,q}^2] - \mathrm{E}[Z_{d,q}S_{d,q}]^2\right)\mathbf{1}_{N\times N}$. The operator $\odot$ refers to an element-wise product, and it is also known as the Hadamard product. While, $\psi(\cdot)$ and $\Gamma(\cdot)$ are the digamma and gamma function, respectively. It is important to notice that the first six terms of the lower bound defined above have the same form as the lower bound found in Álvarez et al. (2010); Titsias (2009).

To perform variational inference, we obtain the update equations for the posterior distributions $q(\mathbf{u})$, $q(\mathbf{S}, \mathbf{Z})$, $q(\boldsymbol{\gamma})$, and $q(\boldsymbol{\upsilon})$ from the lower bound given above. The update equations are included in appendix A, while the mean and the variance for the predictive distribution $p(\mathbf{y}_*|\mathbf{y}, \boldsymbol{\theta})$ appear on appendix B.

5 Related work

Convolved multiple output Gaussian Processes have been successfully applied to regression tasks, such as motion capture data, gene expression data, sensor network data, among others. In the context of latent force models, multiple output Gaussian processes can be used to probabilistically describe several interconnected dynamical systems, with the advantage that the differential equations that describe those systems, and the data observations, work together for accomplishing system identification (Álvarez et al., 2013). Multiple output Gaussian processes are commonly trained assuming that each output is fully connected to all latent functions, this is, the value of each hidden variable $Z_{d,q}$ is equal to one for all d, and q.
Several methods have been proposed in the literature for the problem of model selection in related areas of multiple output Gaussian processes. For example in multi-task learning, a Bayesian multi-task learning model capable to learn the sparsity pattern of the data features base on matrix-variate Gaussian scale mixtures is proposed in Guo et al. (2011). Later, a multi-task learning algorithm that allows sharing one or more latent basis for task belonging to different groups is presented in Kumar and III (2012). This algorithm is also capable of finding the number of latent basis, but it does not place a matrix variate prior ever the sensitivities.
In a closely related work in multi-task Gaussian processes (Titsias and Lázaro-Gredilla, 2011), the problem of model selection was approached using the spike and slab distribution as prior over the weight matrix of a linear combination of Gaussian processes latent functions. The inference step is performed using the variational approach.

6 Implementation

In this section, we briefly describe the covariance functions used in the experiments. The first type of covariance function is based on a convolution of two exponential functions with squared argument. The second type of covariance functions is based on the solution of ordinary differentials equations (ODE), and each data point is linked to a time value.

6.1 Covariance functions

We use three different types of kernels derived from expressions $k_{f_d^q,f_{d'}^q}(\mathbf{x},\mathbf{x}')$ in (2), and $k_{f_d^q,u_q}(\mathbf{x},\mathbf{x}')$ in (3). In turn, these expressions depend of the particular forms for $G_d(\cdot)$, and $k_q(\cdot,\cdot)$.

6.1.1 General purpose covariance function

Here we present a general purpose covariance function for multi-output GPs for which $\mathbf{x} \in \mathcal{X} \in \mathbb{R}^p$. If we assume that both the smoothing kernel $G_d(\cdot)$ and $k_q(\cdot,\cdot)$ have the following form

$$k(\mathbf{x},\mathbf{x}') = \frac{|\mathbf{P}|^{1/2}}{(2\pi)^{p/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mathbf{x}')^\top \mathbf{P}(\mathbf{x}-\mathbf{x}')\right],$$

where $\mathbf{P}$ is a precision matrix, then it can be shown that the covariance function $k_{f_d^q,f_{d'}^q}(\mathbf{x},\mathbf{x}')$ follows as

$$k_{f_d^q,f_{d'}^q}(\mathbf{x},\mathbf{x}') = \frac{1}{(2\pi)^{p/2}|\mathbf{P}_{d,d'}^q|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mathbf{x}')^\top (\mathbf{P}_{d,d'}^q)^{-1}(\mathbf{x}-\mathbf{x}')\right],$$

where $\mathbf{P}_{d,d'}^q = \mathbf{P}_d^{-1} + \mathbf{P}_{d'}^{-1} + \mathbf{\Lambda}_q^{-1}$. Matrices $\mathbf{P}_d$ and $\mathbf{\Lambda}_q$ correspond to the precision matrices associated to $G_d(\cdot)$, and $k_q(\cdot,\cdot)$, respectively. For the experiments, we use diagonal forms for both matrices, where $\{p_{d,i}\}_{i=1}^p$ are the elements for the diagonal matrix $\mathbf{P}_d$, and $\{\ell_{q,i}\}_{i=1}^p$ are the diagonal elements of the matrix $\mathbf{\Lambda}_q$. In the following sections, we refer to this covariance function as the Gaussian Smoothing (GS) kernel.

6.1.2 Latent force models

Latent force models (LFM) can be seen as a hybrid approach that combines differential equations and Gaussian processes (Álvarez et al., 2009; Álvarez et al., 2013). They are built from convolution processes by means of a

deterministic function (which relates the data to a physical model), and Gaussian process priors for the latent functions. In the next two sections we show examples with a first order ordinary differential equation (ODE1), and a second order ordinary differential equation (ODE2). In both cases, we assume the latent functions to be Gaussian processes with zero mean and covariance functions given by

$$k_q(t,t') = \exp\left[-\frac{(t-t')^2}{l_q^2}\right]. \tag{4}$$

We can derive the covariance function for the outputs following (2). In this context, the smoothig kernel $G_d(\cdot)$ is known as the Green's function, and its form will depend on the order of the differential equation.

First order differential equation (ODE1) We assume that the data can be modelled by the first order differential equation given by

$$\frac{\mathrm{d}f_d(t)}{\mathrm{d}t} + B_d f_d(t) = \sum_{q=1}^{Q} S_{d,q} u_q(t), \tag{5}$$

where B_d is the decay rate for output d. Solving for $f_d(t)$ in Equation (5), we get a similar expression to the one obtained in equation (1), where the smoothing kernel $G_d(\cdot)$ (or the Green's function in this context) is given by

$$G_d(t-t') = \exp\left[-B_d(t-t')\right].$$

Using the above form for the smoothing kernel $G_d(\cdot)$, and the covariance function $k_q(\cdot,\cdot)$ given in (4), we derive the expression for $k_{f_d^q, f_{d'}^q}(t,t')$ using (2), which is (Lawrence et al., 2006)

$$k_{f_d^q, f_{d'}^q} = \frac{\sqrt{\pi} l_q}{2}[h_q(B_{d'}, B_d, t', t) + h_q(B_d, B_{d'}, t, t')],$$

where $h_q(B_{d'}, B_d, t', t)$ is defined as

$$\begin{aligned} h_q(B_{d'}, B_d, t', t) =& \frac{\exp(\nu_{q,d'}^2)}{B_d + B_{d'}} \exp(-B_{d'}t') \Bigg\{ \exp(B_{d'}t) \left[\operatorname{erf}\left(\frac{t'-t}{l_q} - \nu_{q,d'}\right) + \operatorname{erf}\left(\frac{t}{l_q} + \nu_{q,d'}\right)\right] \\ & - \exp(-B_d t) \left[\operatorname{erf}\left(\frac{t'}{l_q} - \nu_{q,d'}\right) + \operatorname{erf}(\nu_{q,d'})\right] \Bigg\}, \end{aligned} \tag{6}$$

with $\nu_{q,d} = l_q B_d/2$, and $\operatorname{erf}(\cdot)$ is the error function defined as

$$\operatorname{erf}(z) = \frac{1}{2\pi} \int_0^z \exp\left(-t^2\right) \mathrm{d}t.$$

In order to infer the latent functions $u_q(t)$ related in (5), we calculate the cross-covariance function between $f_d(t)$ and $u_q(t)$ using (3), as follows

$$k_{f_d,u_q}(t,t') = S_{d,q} \frac{\sqrt{\pi} l_q}{2} \exp(\nu_{q,d}^2) \exp\left[-B_d(t-t')\right] \left[\operatorname{erf}\left(\frac{t-t'}{l_q} - \nu_{q,d}\right) + \operatorname{erf}\left(\frac{t'}{l_q} + \nu_{q,d}\right)\right].$$

Second order differential equation (ODE2) In this scenario, we assume that the data can be explained using a second order differential equation related to a mechanical system

$$m_d \frac{\mathrm{d}^2 f_d(t)}{\mathrm{d}t^2} + C_d \frac{\mathrm{d}f_d(t)}{\mathrm{d}t} + B_d f_d(t) = \sum_{q=1}^{Q} S_{d,q} u_q(t), \tag{7}$$

where $\{m_d\}_{d=1}^D$ are mass constants, $\{C_d\}_{d=1}^D$ are damper constants, and $\{B_d\}_{d=1}^D$ are spring constants. Without loss of generality, the value of the mass m_d is set to one. Now, assuming initial conditions equal to zero, the solution for the Green's function associated to (7) is given by

$$G_d(t-t') = \frac{1}{\omega_d} \exp\left[-\alpha_d(t-t')\right] \sin\left[\omega_d(t-t')\right],$$

where α_d is the decay rate and ω_d is the natural frequency. Both variables are defined as

$$\alpha_d = \frac{C_d}{2}, \qquad \omega_d = \frac{\sqrt{4B_d - C_d^2}}{2}.$$

It can be shown that $k_{f_d^q, f_{d'}^q}(t, t')$ reduces to (Álvarez et al., 2009)

$$\begin{aligned} k_{f_d^q, f_{d'}^q}(t, t') =& K_0[h_q(\tilde{\gamma}_{d'}, \gamma_d, t, t') + h_q(\gamma_d, \tilde{\gamma}_{d'}, t', t) + h_q(\gamma_{d'}, \tilde{\gamma}_d, t, t') + h_q(\tilde{\gamma}_d, \gamma_{d'}, t', t) \\ & - h_q(\tilde{\gamma}_{d'}, \tilde{\gamma}_d, t, t') - h_q(\tilde{\gamma}_d, \tilde{\gamma}_{d'}, t', t) - h_q(\tilde{\gamma}_{d'}, \tilde{\gamma}_d, t, t') - h_q(\gamma_d, \gamma_{d'}, t', t)], \end{aligned}$$

where $K_0 = \frac{l_q\sqrt{\pi}}{8\omega_d\omega_{d'}}$ $\gamma_d = \alpha_d + j\omega_d$, $\tilde{\gamma}_d = \alpha_d - j\omega_d$ and $h_q(\cdot)$ is the function defined in (6). Additionally, if ω_d and $\omega_{d'}$ take real values, the expression above simplifies as

$$k_{f_d^q, f_{d'}^q}(t, t') = 2K_0 \operatorname{Re}\left[h_q(\gamma_{d'}, \tilde{\gamma}_d, t, t') + h_q(\tilde{\gamma_d}, \gamma_{d'}, t', t) - h_q(\gamma_{d'}, \gamma_d, t, t') - h_q(\gamma_d, \gamma_{d'}, t', t)\right],$$

where $\operatorname{Re}(\cdot)$ refers to the real part of the argument. For the cross-covariance $k_{f_d, u_q}(t, t')$, it can be shown that the solution for (3) is

$$k_{f_d, u_q}(t, t') = \frac{l_q S_{d,q}\sqrt{\pi}}{j4\omega_d}[\Upsilon_q(\tilde{\gamma}_d, t, t') - \Upsilon_q(\gamma_d, t, t')],$$

where

$$\Upsilon_q(\gamma_d, t, t') = e^{\frac{l_q^2\gamma_d^2}{4}} e^{-\gamma_d(t-t')}\left[\operatorname{erf}\left(\frac{t-t'}{l_q} - \frac{l_q\gamma_d}{2}\right) + \operatorname{erf}\left(\frac{t'}{l_q} + \frac{l_q\gamma_d}{2}\right)\right].$$

6.2 Variational inference procedure

The variational inference procedure can be summarized as follows. We give initial values to the parameters of each variational distribution, and initial values to the parameters of the covariance functions. We also set the values of α, and Q. An iterative process is then performed until a criterion of convergence is fulfilled. At each iteration, we update the moments for each variational distribution as shown in appendix A. Alongside, every ten iterations in the variational inference method, we estimate the parameters $\boldsymbol{\theta}$ of the kernel functions, by maximizing the lower bound F_V using the scaled conjugate gradient method. The derivatives $\frac{\partial F_V}{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}$, $\frac{\partial F_V}{\partial \mathbf{K}_{\mathbf{u},\mathbf{f}}}$ and $\frac{\partial F_V}{\partial \mathbf{K}_{\mathbf{f},\mathbf{f}}}$, are calculated using expressions similar to the ones obtained in Álvarez et al. (2009). We combine those derivatives with the derivatives of $\mathbf{K}_{\mathbf{u},\mathbf{u}}$, $\mathbf{K}_{\mathbf{u},\mathbf{f}}$, and $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ wrt $\boldsymbol{\theta}$. We use the software GPmat (`https://github.com/SheffieldML/GPmat`) to train and test models based on latent force models.

7 Results

In this section, we show results from different datasets, including: artificial data, motion capture data, and gene expression data. For the artificial datasets, we are interested in recovering the known interconnection matrix ($\mathbf{Z}$) between the latent functions and outputs. For the real datasets, we analyse the regression performance of the proposed method under different configurations.

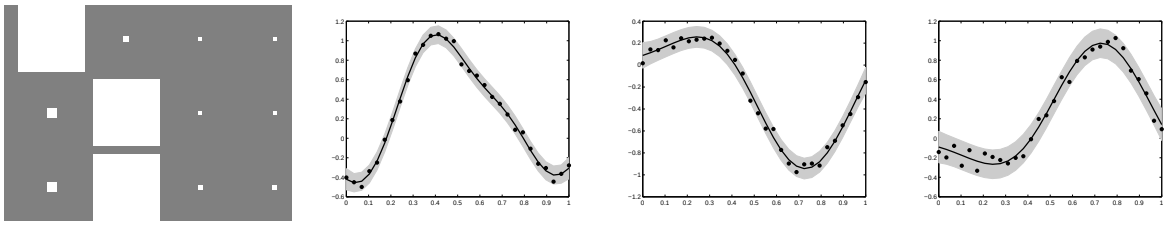
7.1 Synthetic Data

To show the ability of the proposed model to recover the underlying structure between the output data and the latent functions, we apply the method to two different toy multi-output datasets. Each toy dataset is built by sampling from the model explained in section 4.

Example 1: The first experiment is conducted using a GS covariance function (see section 6.1.1) and sample from the model with $D = 3$, $Q = 2$ and $\alpha = 1$. For the smoothing kernels $G_d(x, x')$, we set the length-scales to $p_{1,1} = 0.01$, $p_{2,1} = 1/120$, and $p_{3,1} = 1/140$. We use the following values for matrices $\mathbf{Z}$, and $\mathbf{S}$,

$$\mathbf{Z} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 0 & 1.48 \\ -3.19 & 0 \\ 6.87 & 0 \end{bmatrix}$$

For the covariance functions $k_q(x, x')$ of the latent functions, we choose the length-scales as $l_{1,1} = 0.1$ and $l_{2,1} = 0.2$. Next, we sample the model and generate 30 data points per output, evenly spaced in the interval $[0, 1]$. We assume that each process $w_d(x)$ is a white Gaussian noise process with zero mean, and standard deviation equal to 0.1.
The model is then trained using the proposed variational method with a maximum number of latent functions set to four. Additionally, for the variational distribution of latent functions, we set 15 inducing points evenly space along the output interval.



(a) Hinton diagram toy 1 (b) Ouput 1 toy 1 (c) Ouput 2 toy 1 (d) Ouput 3 toy 1

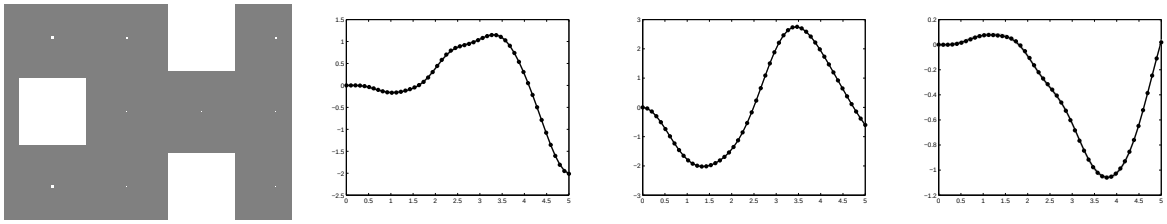
Figure 1: Results for model selection for example 1. Hinton diagram for $\mathbb{E}[Z_{d,q}]$ and, mean and two standard deviations for the predictions over the three outputs.

Figure 1 shows the results of model selection for this experiment. We use a Hinton diagram to display the estimated value for $\mathbb{E}[\mathbf{Z}]$, in Figure 1a. We notice from the Hinton diagram that there are two main latent functions which are used by the model to explain the data. The first column of the Hinton diagram corresponds to the second column of matrix $\mathbf{Z}$, while the second column of Hinton diagram corresponds to the first column of matrix $\mathbf{Z}$. The posterior mean functions for each output closely approximate the data, as shown in Figures 1b to 1d.

Example 2: The second experiment is conducted using an ODE2 covariance function (Álvarez et al., 2009). We generate data using $D = 3$, $Q = 2$ and $\alpha = 1$. For each differential equation, we have the following values for the springs: $B_1 = 4$, $B_2 = 1$, and $B_3 = 1$. The values for the dampers are $C_1 = 0.5$, $C_2 = 4$, and $C_3 = 1$. Matrices $\mathbf{Z}$, and $\mathbf{S}$ are set to the following values

$$\mathbf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} -2.61 & 0 \\ 0 & 2.66 \\ 1.10 & 0 \end{bmatrix}$$

The length-scales for the covariance functions of the latent Gaussian processes were set to $l_{1,1} = 0.2$, and $l_{2,1} = 0.4$. We sample from the model, and generate 50 data points per output evenly spaced across the interval [0,5]. We truncate the number of latent functions to $Q = 4$.



(a) Hinton diagram toy 2 (b) Output 1 toy 2 (c) Output 2 toy 2 (d) Output 3 toy 2

Figure 2: Results for model selection in toy example 2. In Figure 2a, Hinton diagram for $\mathbb{E}[Z_{d,q}]$. In Figures 2b to 2d, mean and two standard deviations for the predictive distribution over the three outputs.

We perform the same evaluation as the one performed in example 1. The Hinton diagram in Figure 2a, shows the values for $\mathbb{E}[Z_{d,q}]$. We recover the structure imposed over the original matrix $\mathbf{Z}$: columns first and third in

the Hinton diagram recover the ones and zeros in $\mathbf{Z}$, whereas columns second and fourth have entries with very small values.
For this experiment, we used $y_d(t) = f_d(t)$ (we did not use an independent process $w_d(t)$). The mean predictive function together with the actual data is shown in Figures 2b to 2d.
In the following sections, we evaluate the performance of the proposed model selection method in human motion capture data and gene expression data.

7.2 Human motion capture data

In this section, we evaluate the performance of the proposed method compared to the Deterministic Training Conditional Variational (DTCVAR) inference procedure proposed in Álvarez et al. (2009). DTCVAR also uses inducing variables for reducing computational complexity within a variational framework, but assumes full connectivity between the latent functions and the output functions (meaning that $Z_{d,q} = 1$, for all q, and d). Parameters for all the kernel functions employed are learned using scaled conjugate gradient optimization.

7.2.1 Performance of the model in terms of changing number of outputs and the truncation level

We use the Carnegie Mellon University's Graphics Lab motion-capture motion capture database available at http://mocap.cs.cmu.edu. Specifically, we consider the movement walking from subject 35 (motion 01). From this movement, we select 20 channels from the 62 available (we avoid channels where the signals were just noise or a straight line). Then, we take 45 frames for training and the rest 313 frames are left out for testing. Table 1 shows a performance comparison between our proposed model and a model trained using DTCVAR, taking into account different types of covariance functions. Performance is measured using standardized mean square error (SMSE) and mean standardized log loss (MSLL) over the test set for different combinations of number of outputs (D) and number of latent functions (Q). Similar results are obtained by both models using the GS covariance function. Even-though, the model based on DTCVAR outperforms the proposed one in the cases $D = 10, Q = 7$, and $D = 15, Q = 9$, the DTCVAR approach uses all latent functions, while the IBP approach uses only two latent functions, as shown in Figure 3.

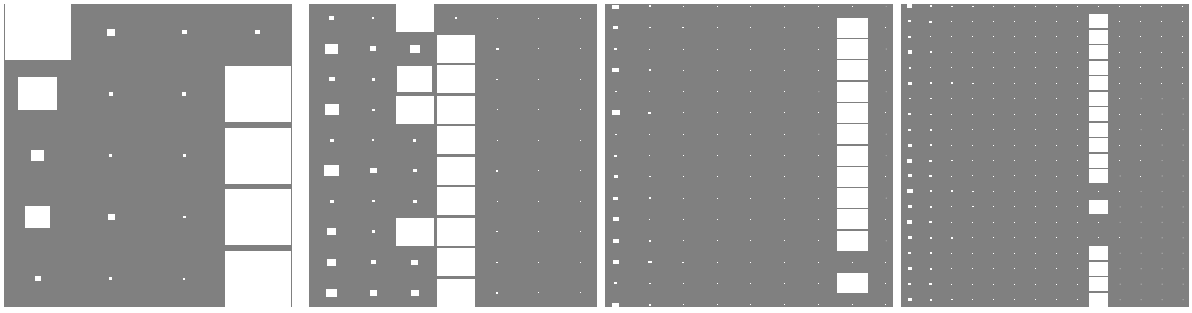
Inference setup	Measure	D=5, Q=4	D=10, Q=7	D=15, Q=9	D=20, Q=14
GS	SMSE	0.241	0.1186	0.3072	0.3796
	MSLL	-1.1334	-1.6924	-1.0322	-0.9196
IBP + GS	SMSE	0.1538	0.3267	0.3494	0.3605
	MSLL	-1.6083	-1.0140	-0.8819	-0.8118

Table 1: Standardized mean square error (SMSE) and mean standardized log loss (MSLL) for different number of outputs and latent functions.

For the other two cases ($D = 5, Q = 4$, and $D = 20, Q = 14$), the proposed method presents a similar performance compared to the model estimated by DTCVAR using all latent functions, showing that to obtain an adequate approximation of the output data we do not require to use the maximum number of the latent functions.

7.2.2 Performance for different kernel functions

In this section, we compare the performance of the proposed model and the one trained using DTCVAR with different covariance functions over the same dataset. In this case, we consider the walking movement from subject 02 motion 01. From the 62 channels, we select 15 for this experiment. We assume a maximum of nine latent functions, and make a comparison between the GS and the ODE2 covariance functions. The latter is used because human motion data consists of recordings of an skeleton's joint angles across time, which summarize the motion. We can use a set of second order differential equations to describe such motion. Table 2 reports the SMSE and MSLL measures for each type of training method and covariance function. Our proposed method presents better results, with the ODE2 kernel being the kernel that best explains the data.



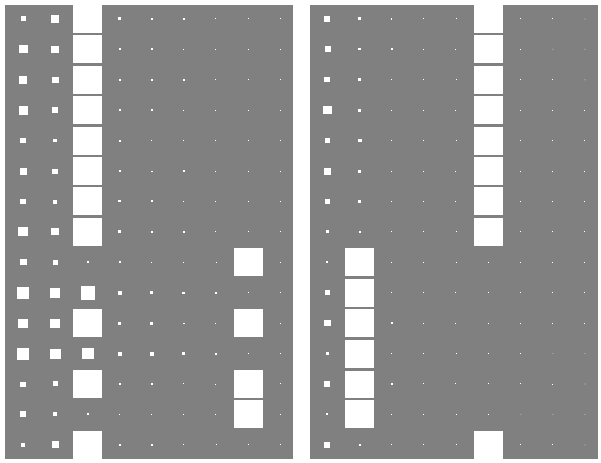
(a) D=5, Q=4 (b) D=10, Q=7 (c) D=15, Q=9 (d) D=20, Q=14

Figure 3: Hinton diagrams of $\mathbb{E}[\mathbf{Z}]$ for each pair of outputs and latent functions tested in table 1.

	ODE2	IBP + ODE2	GS	IBP + GS
SMSE	0.5463	0.2087	0.5418	**0.1790**
SMLL	-0.6547	**-1.2725**	-0.7863	-1.1993

Table 2: Standardized mean square error (SMSE) and mean standardized log loss (MSLL) for different models and different kernel functions.

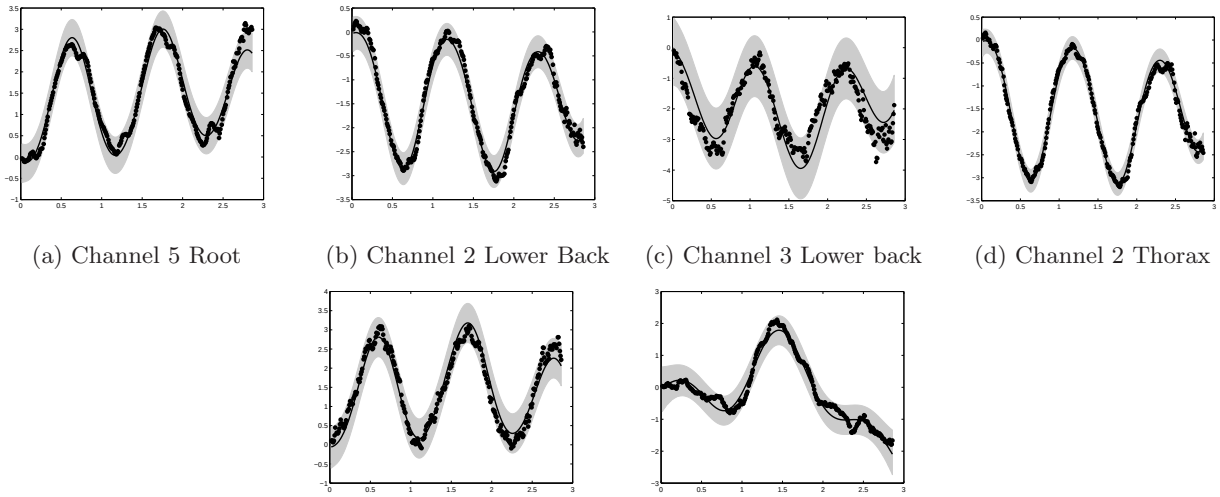
Comparing the Hinton diagrams from both covariance functions (see Figure 4), we find similar results. For example, there is a similar composition of elements between columns 3 and 8 from the Hinton diagram of the ODE2 kernel, with columns 6 and 2 from the Hinton diagram of the GS kernel. Both covariance functions try to unveil a similar interconnection between outputs and latent functions.



(a) IBP + ODE2 (b) IBP + GS

Figure 4: Hinton diagrams for the IBP variational approximation using (a) ODE2 and (b) GS covariance functions.

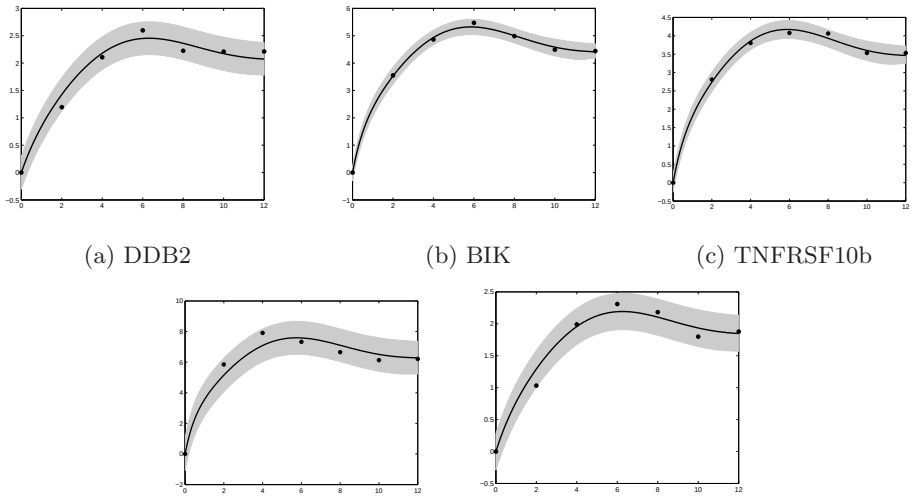
Figure 5 shows the Gaussian process mean and variance for the predictive distribution of six outputs from the model inferred from IBP + ODE2. In most of the predictions, the model explains the testing data points with adequate accuracy.



(a) Channel 5 Root (b) Channel 2 Lower Back (c) Channel 3 Lower back (d) Channel 2 Thorax

(e) Channel 3 Thorax (f) Channel 2 Head

Figure 5: Mean (solid line) and two standard deviations (gray shade) for predictions over six selected outputs from IBP + ODE2 trained model.



(a) DDB2 (b) BIK (c) TNFRSF10b

(d) CIp1/p21 (e) SESN1/hPA26

Figure 6: Mean (solid line) and two standard deviations (grey shade) for the predictions over five gene expression levels from the IBP + ODE1 model.

7.3 Gene expression data: Tumour Suppressor Protein p53

Gene expression data consist of measurements of the mRNA concentration of a set of genes. mRNA concentration for each gene is regulated by the so called transcription factor (TFs) proteins. In transcriptional regulatory networks, a TF or a set of TFs may act in a individually or collaborative manner, leading to complex regulatory interactions.

Gene expression data can be related to a first order differential equation (Barenco et al., 2006), with the same form given in equation 5. From this equation, and in the context of gene expression, the output $f_d(t)$ is the mRNA concentration of gene d, B_d is the linear degradation rate of $f_d(t)$, and $u(t)$ is the concentration of the

TF. Two major problems arise from the analysis of gene expression, first to determine the interaction network and second to infer the activated transcription factor (Gao et al., 2008).
In this experiment, we use tumour suppressor protein p53 dataset from Barenco et al. (2006). This dataset is restricted to five known target genes: DDB2, BIK, TNFRSF10b, CIp1/p21 and SESN1/hPA26. In Lawrence et al. (2006), a transcription factor is inferred from the expression levels of these five target genes using a covariance function build from a first order differential equation. Our aim is to determine the number of transcription factors (latent functions) and how they explain the activities of the target genes using the covariance function defined in 6.1.2.

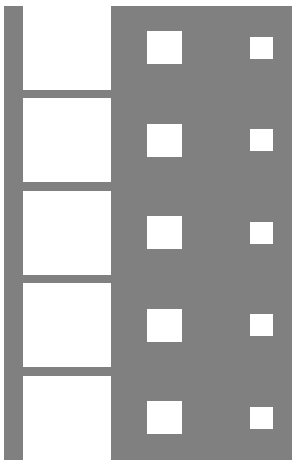


Figure 7: Hinton diagrams for the IBP variational approximation using ODE1 covariance function.

Results obtained from this dataset regarding the number of latent forces, concurred with the description given in Barenco et al. (2006), where there is one protein influencing the expression level of the genes analysed (see Figure 7).

8 Conclusions

We have introduced a new variational method to perform model selection in convolved multiple output Gaussian Processes. Our main aim was to identify the relationship between the latent functions and the outputs in multiple output Gaussian processes. The proposed method achieved comparable results to the model that assumes full connectivity between latent functions and output functions. This makes our method suitable to applications where the complexity of the model should be reduced. The proposed model selection method can be applied in other applications that involve the use of a covariance function based on differential equations, such as inferring the biological network in gene expression microarray data.
For the artificial dataset examples we found that the model selection method converges to a similar matrix for the interconnections between latent functions and outputs. We have illustrated the performance of the proposed methodology for regression of human motion capture data, and a small gene expression dataset.

Acknowledgments

CDG would like to thank to Convocatoria 567 of Colciencias. MAA would like to thank to Banco Santander for the support received under the program "Scholarship for Young Professors and Researchers Iberoamérica". MAA would also like to acknowledge the support from British Council and Colciencias under the research project "Sparse Latent Force Models for Reverse Engineering of Multiple Transcription Factors". The authors of this manuscript would like to thank to Professor Fernando Quintana from Pontificia Universidad Católica de Chile, for his valuable discussions and feedback on this manuscript.

References

Mauricio A. Álvarez and Neil D. Lawrence. Computationally efficient convolved multiple output gaussian processes. *Journal of Machine Learning Research*, 12:1459–1500, 2011.

Mauricio A. Álvarez, David Luengo, and Neil D. Lawrence. Latent Force Models. In David van Dyk and Max Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, pages 9–16, Clearwater Beach, Florida, 16-18 April 2009. JMLR W&CP 5.

Mauricio A. Álvarez, David Luengo, Michalis K. Titsias, and Neil D. Lawrence. Variational inducing kernels for sparse convolved multiple output gaussian processes. Technical report, University of Manchester, 2009.

Mauricio A. Álvarez, David Luengo, Michalis K. Titsias, and Neil D. Lawrence. Efficient multioutput Gaussian processes through variational inducing kernels. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 25–32, Chia Laguna, Sardinia, Italy, 13-15 May 2010. JMLR W&CP 9.

Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: a review. *Foundations and Trends® in Machine Learning*, 4(3):195–266, 2012.

Mauricio A. Álvarez, David Luengo, and Neil D. Lawrence. Linear latent force models using gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2693–2705, 2013.

M. Barenco, D. Tomescu, D. Brewer, R. Callard, J. Stark, and M. Hubank. Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biology*, 7(3):R25, 2006.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Edwin V. Bonilla, Kian Ming Chai, and Christopher K. I. Williams. Multi-task Gaussian process prediction. In John C. Platt, Daphne Koller, Yoram Singer, and Sam Roweis, editors, *NIPS*, volume 20, Cambridge, MA, 2008. MIT Press.

Jean Paul Chilès and Pierre Delfiner. *Geostatistics: Modeling Spatial Uncertainty*. John Wiley & Sons, New York, 1999.

Noel A. C. Cressie. *Statistics for Spatial Data*. John Wiley & Sons (Revised edition), USA, 1993.

Finale Doshi-Velez, Kurt Miller, Jurgen Van Gael, and Yee Whye Teh. Variational inference for the Indian Buffet process. In *AISTATS 2009*, pages 137–144, 2009.

Pei Gao, Michalis K. Titsias, Neil D. Lawrence, and Magnus Rattray. Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities. *Bioinformatics*, 24:70–75, 2008.

Pierre Goovaerts. *Geostatistics For Natural Resources Evaluation*. Oxford University Press, USA, 1997.

Thomas L. Griffiths and Zoubin Ghahramani. Infinite latent feature models and the indian buffet process. In *In NIPS*, pages 475–482. MIT Press, 2005.

Thomas L. Griffiths and Zoubin Ghahramani. The indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12:1185–1224, July 2011. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1953048.2021039`.

Shengbo Guo, Onno Zoeter, and Cédric Archambeau. Sparse bayesian multi-task learning. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1755–1763. Curran Associates, Inc., 2011. URL `http://papers.nips.cc/paper/4242-sparse-bayesian-multi-task-learning.pdf`.

David M. Higdon. A process-convolution approach to modeling temperatures in the north atlantic ocean. *Journal of Ecological and Environmental Statistics*, 5:173–190, 1998.

Andre G. Journel and Charles J. Huijbregts. *Mining Geostatistics*. Academic Press, London, 1978. ISBN 0-12391-050-1.

David A. Knowles and Zoubin Ghahramani. Nonparametric Bayesian sparse factor models with application to gene expression modelling. *Annals of Applied Statistics*, 5(2B):1534–1552, 2011.

Abhishek Kumar and Hal Daumé III. Learning task grouping and overlap in multi-task learning. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, 2012. URL `http://icml.cc/discuss/2012/690.html`.

Neil D. Lawrence, Guido Sanguinetti, and Magnus Rattray. Modelling transcriptional regulation using Gaussian Processes. In *Neural Information Processing Systems*, pages 785–792, 2006.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006. ISBN 0-262-18253-X.

Grigorios Skolidis and Guido Sanguinetti. Bayesian multitask classification with Gaussian process priors. *IEEE*

Transactions on Neural Networks, 22(12):2011 – 2021, 2011.

Michalis K. Titsias. Variational learning of inducing variables in sparse gaussian processes. In *In Artificial Intelligence and Statistics 12*, pages 567–574, 2009.

Michalis K. Titsias and Miguel Lázaro-Gredilla. Spike and slab variational inference for multi-task and multiple kernel learning. In *NIPS 2011*, pages 2339–2347, 2011.

Jay M. Ver Hoef and Ronald Paul Barry. Constructing and fitting models for cokriging and multivariable spatial prediction. *Journal of Statistical Plannig and Inference*, 69:275–294, 1998.

A Computing the optimal posterior distributions

In this appendix, we present the updates of variational distributions $q(\mathbf{S}, \mathbf{Z})$, $q(\mathbf{u})$, $q(\gamma)$ and $q(\upsilon)$. To do so, first, we rewrite the lower bound defined in section 4, as

$$
\begin{aligned}
F_V = &\sum_{q=1}^{Q} \operatorname{tr}\left(\mathbf{m}_q \operatorname{E}[\mathbf{u}_q^\top]\right) - \frac{1}{2}\sum_{q=1}^{Q}\sum_{q'=1}^{Q} \operatorname{tr}\left(\mathbf{P}_{q,q'} \operatorname{E}[\mathbf{u}_{q'}\mathbf{u}_q^\top]\right) - \frac{1}{2}\sum_{q=1}^{Q} \operatorname{tr}\left(\mathbf{K}_{\mathbf{u}_q,\mathbf{u}_q}^{-1} \operatorname{E}[\mathbf{u}_q\mathbf{u}_q^\top]\right) - \frac{QM}{2}\log 2\pi \\
&- \frac{1}{2}\sum_{q=1}^{Q}\log|\mathbf{K}_{\mathbf{u}_q,\mathbf{u}_q}| + \operatorname{H}(\mathbf{u}) - \frac{ND}{2}\log 2\pi - \frac{1}{2}\sum_{d=1}^{D}\log|\mathbf{\Sigma}_{\mathbf{w}_d}| - \frac{1}{2}\sum_{d=1}^{D}\operatorname{tr}\left(\mathbf{\Sigma}_{\mathbf{w}_d}^{-1}\mathbf{y}_d\mathbf{y}_d^\top\right) - \frac{1}{2}\log 2\pi\sum_{d=1}^{D}\sum_{q=1}^{Q}\operatorname{E}[Z_{d,q}] \\
&+ \sum_{d=1}^{D}\sum_{q=1}^{Q}\operatorname{E}[Z_{d,q}]\operatorname{E}[\log\pi_q] + \frac{1}{2}\sum_{d=1}^{D}\sum_{q=1}^{Q}\operatorname{E}[Z_{d,q}]\left[\psi(a_{d,q}^{\gamma^*}) - \log b_{d,q}^{\gamma^*}\right] - \frac{1}{2}\sum_{d=1}^{D}\sum_{q=1}^{Q}\left(\frac{a_{d,q}^{\gamma^*}}{b_{d,q}^{\gamma^*}} + c_{d,q}\right)\operatorname{E}[Z_{d,q}S_{d,q}^2] \\
&+ \sum_{d=1}^{D}\sum_{q=1}^{Q}(1-\operatorname{E}[Z_{d,q}])\operatorname{E}[\log(1-\pi_q)] - \sum_{d=1}^{D}\sum_{q=1}^{Q}\log\Gamma(a_{d,q}^{\gamma}) + \sum_{d=1}^{D}\sum_{q=1}^{Q}a_{d,q}^{\gamma}\log b_{d,q}^{\gamma} \\
&+ \sum_{d=1}^{D}\sum_{q=1}^{Q}\left(a_{d,q}^{\gamma}-1\right)\left[\psi(a_{d,q}^{\gamma^*}) - \log b_{d,q}^{\gamma^*}\right] - \sum_{d=1}^{D}\sum_{q=1}^{Q}b_{d,q}^{\gamma}\frac{a_{d,q}^{\gamma^*}}{b_{d,q}^{\gamma^*}} + (\alpha-1)\sum_{q=1}^{Q}\left[\psi(\tau_{q1}) - \psi(\tau_{q1}+\tau_{q2})\right] + Q\log\alpha \\
&+ \operatorname{H}(\mathbf{S},\mathbf{Z}) + \operatorname{H}(\boldsymbol{\upsilon}) + \operatorname{H}(\boldsymbol{\gamma}),
\end{aligned}
$$

with

$$
\begin{aligned}
c_{d,q} =& \operatorname{tr}\left(\mathbf{\Sigma}_{\mathbf{w}_d}^{-1}\mathbf{K}_{\mathbf{f}_d|\mathbf{u}_q}\right), \\
\mathbf{K}_{\mathbf{f}_d|\mathbf{u}_q} =& \mathbf{K}_{\mathbf{f}_d\mathbf{f}_d}^{(q)} - \mathbf{K}_{\mathbf{f}_d\mathbf{u}_q}\mathbf{K}_{\mathbf{u}_q\mathbf{u}_q}^{-1}\mathbf{K}_{\mathbf{f}_d,\mathbf{u}_q}^\top, \\
\mathbf{m}_q =& \sum_{d=1}^{D}\operatorname{E}[Z_{d,q}S_{d,q}]\mathbf{K}_{\mathbf{u}_q,\mathbf{u}_q}^{-1}\mathbf{K}_{\mathbf{f}_d,\mathbf{u}_q}^\top\mathbf{\Sigma}_{\mathbf{w}_d}^{-1}\mathbf{y}_d, \\
\mathbf{P}_{q,q'} =& \sum_{d=1}^{D}\operatorname{E}\left[Z_{d,q}S_{d,q}Z_{d,q'}S_{d,q'}\right]\mathbf{K}_{\mathbf{u}_q,\mathbf{u}_q}^{-1}\mathbf{K}_{\mathbf{f}_d,\mathbf{u}_q}^\top\mathbf{\Sigma}_{\mathbf{w}_d}^{-1}\mathbf{K}_{\mathbf{f}_d,\mathbf{u}_{q'}}\mathbf{K}_{\mathbf{u}_{q'},\mathbf{u}_{q'}}^{-1}.
\end{aligned}
$$

Additionally, $\mathbf{\Sigma}_{\mathbf{w}}$ is the covariance matrix for the observation noise.

A.1 Updates for distribution $q(\mathbf{u})$

Taking into account that $q(\mathbf{u}) = \prod_{q=1}^{Q} q(\mathbf{u}_q)$ and $q(\mathbf{u}_q) = \mathcal{N}(\mathbf{u}_q|\widetilde{\mathbf{u}}_q, \widetilde{\mathbf{K}}_{\mathbf{u}_q,\mathbf{u}_q})$. Then, it can be shown that the moment updates are

$$
\widetilde{\mathbf{K}}_{\mathbf{u}_i,\mathbf{u}_i}^{-1} = \mathbf{P}_{i,i} + \mathbf{K}_{\mathbf{u}_i,\mathbf{u}_i}^{-1} \quad \widetilde{\mathbf{u}}_q = \widetilde{\mathbf{K}}_{\mathbf{u}_i,\mathbf{u}_i}\widehat{\mathbf{u}}_i,
$$

with

$$
\widehat{\mathbf{u}}_i = \sum_{d=1}^{D}\operatorname{E}[Z_{d,i}S_{d,i}]\mathbf{K}_{\mathbf{u}_i,\mathbf{u}_i}^{-1}\mathbf{K}_{\mathbf{f}_d,\mathbf{u}_i}^\top\mathbf{\Sigma}_{\mathbf{w}_d}^{-1}\left(\mathbf{y}_d - \widehat{\mathbf{y}}\right),
$$

and

$$\widehat{\mathbf{y}} = \sum_{q'=1, q' \neq i}^{Q} \mathrm{E}\left[Z_{d,q'} S_{d,q'}\right] \mathbf{K}_{\mathbf{f}_d, \mathbf{u}_{q'}} \mathbf{K}_{\mathbf{u}_{q'}, \mathbf{u}_{q'}}^{-1} \mathrm{E}[\mathbf{u}_{q'}].$$

A.2 Updates for distribution $q(\mathbf{S}, \mathbf{Z})$

We assume that the distribution $q(\mathbf{S}, \mathbf{Z})$ is defined as

$$q(\mathbf{S}, \mathbf{Z}) = q(\mathbf{S}|\mathbf{Z}) q(\mathbf{Z}) = \prod_{d=1}^{D} \prod_{q=1}^{Q} q(S_{d,q}|Z_{d,q}) q(Z_{d,q}).$$

First, we calculate the update parameters for the distribution $q(Z_{d,q})$, which is defined as

$$q(Z_{d,q}) = \eta_{d,q}^{Z_{d,q}} (1 - \eta_{d,q})^{1 - Z_{d,q}}.$$

Also, $q(S_{d,q}|Z_{d,q})$ is defined as

$$q(S_{d,q}|Z_{d,q} = 1) = \mathcal{N}(S_{d,q}|\mu_{S_{d,q}}, \nu_{d,q}).$$

Then it can be shown that the update for $\nu_{d,i}$ is given by

$$\nu_{d,i}^{*} = \left(\mathrm{tr}\left(\mathbf{P}_{d,i,i} \left[\widetilde{\mathbf{K}}_{\mathbf{u}_i, \mathbf{u}_i} + \mathrm{E}[\mathbf{u}_i] \, \mathrm{E}[\mathbf{u}_i]^{\top} \right] \right) + \frac{a_{d,i}^{\gamma^*}}{b_{d,i}^{\gamma^*}} + c_{d,i} \right)^{-1},$$

with

$$\mathbf{P}_{d,q,q'} = \mathbf{K}_{\mathbf{u}_q, \mathbf{u}_q}^{-1} \mathbf{K}_{\mathbf{f}_d, \mathbf{u}_q}^{\top} \mathbf{\Sigma}_{\mathbf{w}_d}^{-1} \mathbf{K}_{\mathbf{f}_d, \mathbf{u}_{q'}} \mathbf{K}_{\mathbf{u}_{q'}, \mathbf{u}_{q'}}^{-1}.$$

While, the update for parameter $\mu_{S_{d,i}}$ is calculated as

$$\mu_{S_{d,i}} = \nu_{d,i}^{*} \left[\mathrm{tr}\left(\mathbf{m}_{d,i} \, \mathrm{E}[\mathbf{u}_i^{\top}] \right) - \sum_{q'=1, q' \neq i}^{Q} \mathrm{tr}\left(\eta_{d,q'} \mu_{S_{d,q'}} \mathbf{P}_{d,i,q'} \, \mathrm{E}[\mathbf{u}_{q'}] \, \mathrm{E}[\mathbf{u}_i^{\top}] \right) \right],$$

with

$$\mathbf{m}_{d,q} = \mathbf{K}_{\mathbf{u}_q, \mathbf{u}_q}^{-1} \mathbf{K}_{\mathbf{f}_d, \mathbf{u}_q}^{\top} \mathbf{\Sigma}_{\mathbf{w}_d}^{-1} \mathbf{y}_d.$$

Finally, the update for $\eta_{d,i}$ is given by

$$\begin{aligned} \ln \frac{\eta_{d,i}}{1 - \eta_{d,i}} = \vartheta_{d,i} = {} & \mathrm{tr}\left(\mu_{S_{d,i}} \mathbf{m}_{d,i} \, \mathrm{E}[\mathbf{u}_i^{\top}] \right) - \sum_{q'=1, q' \neq i}^{Q} \mathrm{tr}\left(\mu_{S_{d,i}} \eta_{d,q'} \mu_{S_{d,q'}} \mathbf{P}_{d,i,q'} \, \mathrm{E}[\mathbf{u}_{q'} \mathbf{u}_i^{\top}] \right) \\ & - \frac{1}{2} \mathrm{tr}\left(\left(\nu_{d,i} + \mu_{S_{d,i}}^2 \right) \mathbf{P}_{d,i,i} \left[\widetilde{\mathbf{K}}_{\mathbf{u}_i, \mathbf{u}_i} + \mathrm{E}[\mathbf{u}_i] \, \mathrm{E}[\mathbf{u}_i]^{\top} \right] \right) - \frac{1}{2} \log 2\pi + \mathrm{E}[\log \pi_i] \\ & + \frac{1}{2} \left[\psi(a_{d,i}^{\gamma^*}) - \log b_{d,i}^{\gamma^*} \right] - \frac{1}{2} \left(\frac{a_{d,i}^{\gamma^*}}{b_{d,i}^{\gamma^*}} + c_{d,i} \right) \left(\nu_{d,i} + \mu_{S_{d,i}}^2 \right) - \mathrm{E}[\log(1 - \pi_i)] + \frac{1}{2} \ln(2\pi e^1 \nu_{d,i}), \end{aligned}$$

where

$$\eta_{d,i} = \frac{1}{1 + e^{-\vartheta_{d,i}}}, \quad \mathrm{E}_{q(\boldsymbol{\upsilon})}[\log \pi_q] = \sum_{i=1}^{q} [\psi(\tau_{i1}) - \psi(\tau_{i1} + \tau_{i2})].$$

For computing $\mathrm{E}_{q(\boldsymbol{\upsilon})}[\log(1 - \prod_{i=1}^{q} \upsilon_i)]$, we would need to resort to a local variational approximation (Bishop, 2006) in a similar way to Doshi-Velez et al. (2009).

A.3 Updates for distribution $q(\boldsymbol{\gamma})$

This distribution is defined as

$$q(\boldsymbol{\gamma}) = \prod_{d=1}^{D}\prod_{q=1}^{Q} \text{Gamma}(\gamma_{d,q}|a_{d,q}^{\gamma^*}, b_{d,q}^{\gamma^*}).$$

It can be shown that the updates for the parameters $b_{d,q}^{\gamma^*}$ and $a_{d,q}^{\gamma^*}$ are given by

$$b_{d,q}^{\gamma^*} = \frac{1}{2}\,\text{E}[Z_{d,q}S_{d,q}^2] + b_{d,q}^{\gamma},$$

and

$$a_{d,q}^{\gamma^*} = \frac{1}{2}\,\text{E}[Z_{d,q}] + a_{d,q}^{\gamma}.$$

A.4 Updates for distribution $q(\boldsymbol{\upsilon})$

We assumed that the optimal distribution for each $q(\upsilon_i) = \text{Beta}(\upsilon_i|\tau_{i1}, \tau_{i2})$. Given a fixed value for i, the updates for parameters τ_{i1} and τ_{i2} are given by

$$\tau_{i1} = \alpha + \sum_{m=i}^{Q}\sum_{d=1}^{D}\eta_{d,m} + \sum_{m=i+1}^{Q}\left[\sum_{d=1}^{D}(1-\eta_{d,m})\sum_{j=i+1}^{m} q_{mj}\right],$$

$$\tau_{i2} = 1 + \sum_{m=i}^{Q}\sum_{d=1}^{D}(1-\eta_{d,m})q_{mi}.$$

B Predictive distribution

For the predictive distribution, we need to compute

$$p(\mathbf{y}_*|\boldsymbol{\theta}) = \int_{\mathbf{S},\mathbf{Z}}\int_{\mathbf{u},u} p(\mathbf{y}_*|\boldsymbol{\theta}, u, \mathbf{S}, \mathbf{Z})q(u,\mathbf{u})q(\mathbf{S},\mathbf{Z})\text{d}u\;\text{d}\mathbf{u}\;\text{d}\mathbf{S}\;\text{d}\mathbf{Z}.$$

It can be demonstrated that the above integral is intractable. Since we are only interested in the mean and covariance for $\mathbf{y}_*$, we can still compute them using

$$\text{E}[\mathbf{y}_*] = \int_{\mathbf{y}_*}\mathbf{y}_* p(\mathbf{y}_*|\cdot)\text{d}\mathbf{y}_* = \int_{\mathbf{S},\mathbf{Z}}\left[\int_{\mathbf{y}_*}\mathbf{y}_* p(\mathbf{y}_*|\cdot)\text{d}\mathbf{y}_*\right]q(\mathbf{S},\mathbf{Z})\text{d}\mathbf{S}\;\text{d}\mathbf{Z} = \int_{\mathbf{S},\mathbf{Z}}\text{E}_{\mathbf{y}_*|\cdot}[\mathbf{y}_*]q(\mathbf{S},\mathbf{Z})\text{d}\mathbf{S}\;\text{d}\mathbf{Z},$$

we get

$$\text{E}_{\mathbf{y}_d}[\mathbf{y}_d] = \sum_{k=1}^{Q}\text{E}[Z_{d,k}S_{d,k}]\widehat{\boldsymbol{\alpha}}_{d,k}$$

where $\widehat{\boldsymbol{\alpha}}_{d,k} = \mathbf{K}_{\mathbf{f}_d,\mathbf{u}_k}\mathbf{A}_{k,k}^{-1}\left(\sum_{j=1}^{D}\widetilde{\mathbf{K}}_{\mathbf{u}_k,\mathbf{f}_j}\boldsymbol{\Sigma}_{\mathbf{w}_j}^{-1}\widetilde{\mathbf{y}}_j\right)$, and $\widehat{\boldsymbol{\alpha}}_d = [\widehat{\boldsymbol{\alpha}}_{d,1},\ldots,\widehat{\boldsymbol{\alpha}}_{d,Q}]$. Notice that the $\mathbf{K}_{\mathbf{f}_d,\mathbf{u}_k}$ in the expression for $\widehat{\boldsymbol{\alpha}}_{d,k}$ must be computed at the test point $\mathbf{x}_*$.

We need to compute now the second moment of $\mathbf{y}_*$ under $p(\mathbf{y}_*|\cdot)$. Again, we can write

$$\text{E}[\mathbf{y}_*\mathbf{y}_*^\top] = \int_{\mathbf{y}_*}\mathbf{y}_*\mathbf{y}_*^\top p(\mathbf{y}_*|\cdot)\text{d}\mathbf{y}_* = \int_{\mathbf{S},\mathbf{Z}}\left[\int_{\mathbf{y}_*}\mathbf{y}_*\mathbf{y}_*^\top p(\mathbf{y}_*|\cdot)\text{d}\mathbf{y}_*\right]q(\mathbf{S},\mathbf{Z})\text{d}\mathbf{S}\;\text{d}\mathbf{Z} = \int_{\mathbf{S},\mathbf{Z}}\text{E}_{\mathbf{y}_*|\cdot}[\mathbf{y}_*\mathbf{y}_*^\top]q(\mathbf{S},\mathbf{Z})\text{d}\mathbf{S}\;\text{d}\mathbf{Z},$$

where $\text{E}_{\mathbf{y}_*|\cdot}[\mathbf{y}_*\mathbf{y}_*^\top]$ is the second moment of $\mathbf{y}_*$ under the density $p(\mathbf{y}_*|\cdot)$, we get

$$\text{E}_{\mathbf{y}_*}[\mathbf{y}_d\mathbf{y}_d^\top] = \sum_{i=1}^{Q}\text{E}[Z_{d,i}S_{d,i}^2]\mathbf{K}_{\mathbf{f}_d,\mathbf{f}_d}^{(i)} - \sum_{i=1}^{Q}\text{E}[Z_{d,i}S_{d,i}^2]\mathbf{K}_{\mathbf{f}_d,\mathbf{u}_i}\boldsymbol{\Gamma}_{i,i}\mathbf{K}_{\mathbf{u}_i,\mathbf{f}_d} - \sum_{i=1}^{Q}\sum_{j=1}^{Q}\text{E}[Z_{d,i}S_{d,i}Z_{d,j}S_{d,j}]\widehat{\boldsymbol{\alpha}}_{d,i}\widehat{\boldsymbol{\alpha}}_{d,j}^\top + \boldsymbol{\Sigma}_{\mathbf{w}_d}.$$

Finally, the covariance $\text{cov}[\mathbf{y}_d \mathbf{y}_d^\top]$ would be given as

$$\text{cov}[\mathbf{y}_d \mathbf{y}_d^\top] = \sum_{i=1}^{Q} \text{E}[Z_{d,i} S_{d,i}^2] \mathbf{K}^{(i)}_{\mathbf{f}_d, \mathbf{f}_d} - \sum_{i=1}^{Q} \text{E}[Z_{d,i} S_{d,i}^2] \mathbf{K}_{\mathbf{f}_d, \mathbf{u}_i} \mathbf{\Gamma}_{i,i} \mathbf{K}_{\mathbf{u}_i, \mathbf{f}_d} + \mathbf{\Sigma}_{\mathbf{w}_d}.$$

Bear in mind, we have omitted $*$ in the vectors above to keep the notation uncluttered.